Infinite Models II

Zoubin Ghahramani
Center for Automated Learning and Discovery
Carnegie Mellon University
`http://www.cs.cmu.edu/∼zoubin`

Mar 2002

Carl E. Rasmussen
Matthew J. Beal

Gatsby Computational Neuroscience Unit
University College London
`http://www.gatsby.ucl.ac.uk/`

Two conflicting Bayesian views?

View 1: Occam's Razor. Bayesian learning automatically finds the optimal model complexity given the available amount of data, since Occam's Razor is an integral part of Bayes [Jefferys & Berger; MacKay]. Occam's Razor discourages overcomplex models.

View 2: Large models. There is no *statistical* reason to constrain models; use large models (no matter how much data you have) [Neal] and pursue the infinite limit if you can [Neal; Williams, Rasmussen].

Both views require averaging over all model parameters.

These two views seem contradictory.

Example, should we use Occam's Razor to find the "best" number of hidden units in a feedforward neural network, or simply use as many hidden units as we can manage computationally?

View 1: Finding the “best” model complexity

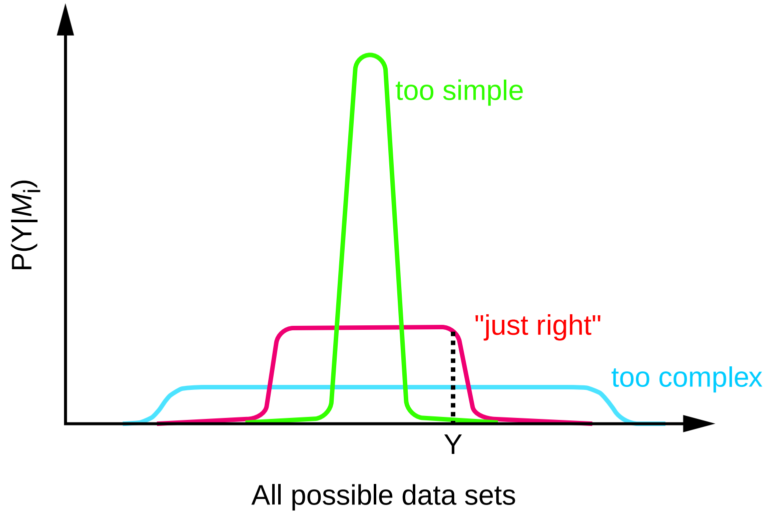
Select the model class with the highest probability given the data:

$$P(\mathcal{M}_i|Y) = \frac{P(Y|\mathcal{M}_i)P(\mathcal{M}_i)}{P(Y)}, \qquad P(Y|\mathcal{M}_i) = \int_{\theta_i} P(Y|\theta_i, \mathcal{M}_i)P(\theta_i|\mathcal{M}_i)\, d\theta_i$$

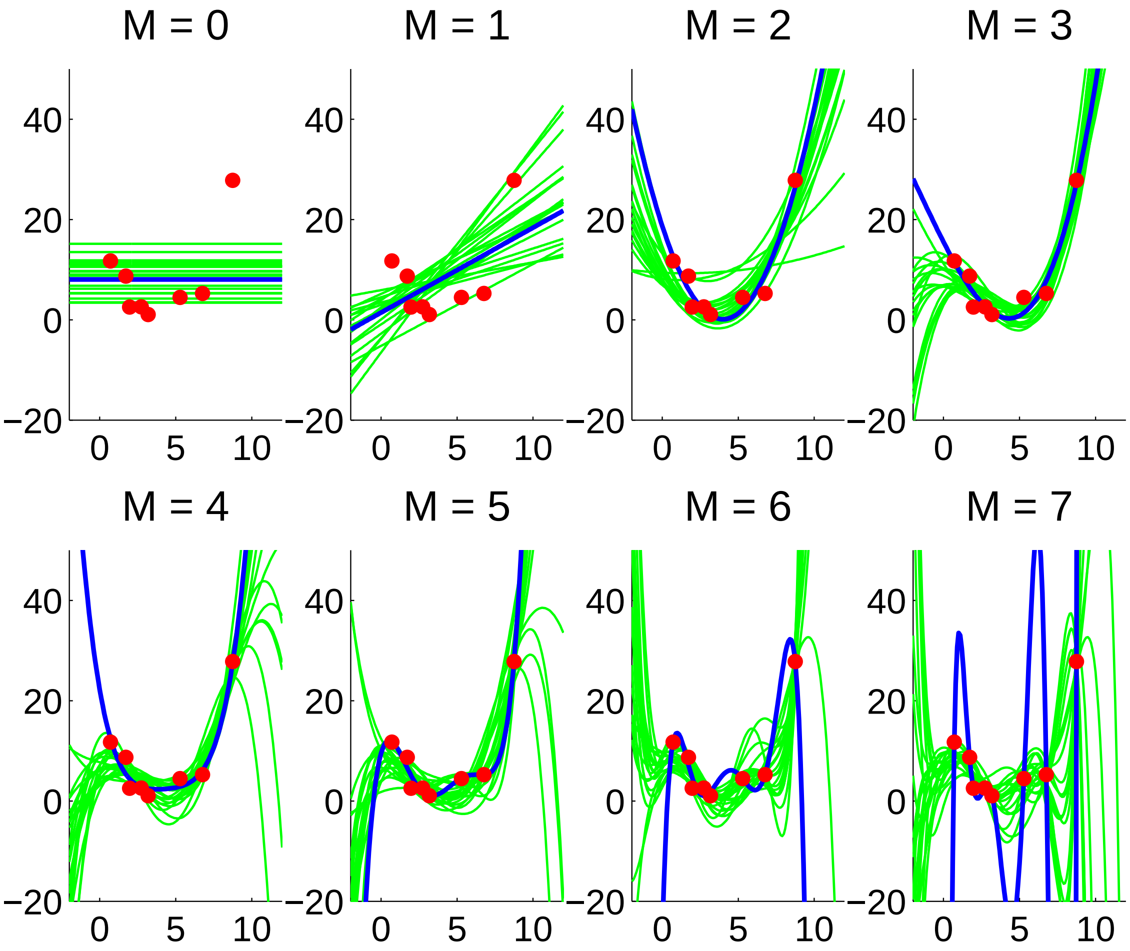
Interpretation: The probability that *randomly selected* parameter values from the model class would generate data set Y.

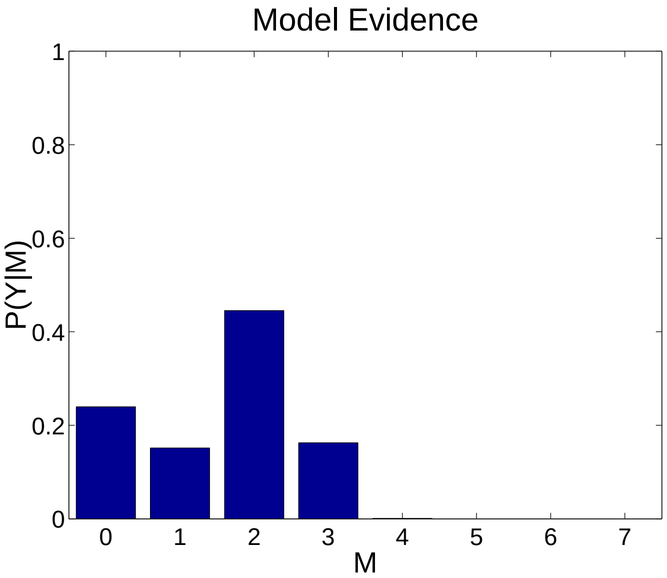
Model classes that are too simple are unlikely to generate the data set.

Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.



Bayesian Model Selection: Occam’s Razor at Work





Lower Bounding the Evidence

Variational Bayesian Learning

Let the hidden states be $\mathbf{x}$, data $\mathbf{y}$ and the parameters $\boldsymbol{\theta}$.
We can lower bound the evidence (Jensen's inequality):

$$\begin{aligned}
\ln P(\mathbf{y}|\mathcal{M}) &= \ln \int d\mathbf{x}\, d\boldsymbol{\theta}\; P(\mathbf{y},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}) \\
&= \ln \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q(\mathbf{x},\boldsymbol{\theta}) \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q(\mathbf{x},\boldsymbol{\theta})} \\
&\geq \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q(\mathbf{x},\boldsymbol{\theta}) \ln \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q(\mathbf{x},\boldsymbol{\theta})}.
\end{aligned}$$

Use a simpler, factorised approximation to $Q(\mathbf{x},\boldsymbol{\theta})$:

$$\begin{aligned}
\ln P(\mathbf{y}) &\geq \int d\mathbf{x}\, d\boldsymbol{\theta}\; Q_{\mathbf{x}}(\mathbf{x}) Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ln \frac{P(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{Q_{\mathbf{x}}(\mathbf{x}) Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \\
&= \mathcal{F}(Q_{\mathbf{x}}(\mathbf{x}), Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \mathbf{y}).
\end{aligned}$$

Variational Bayesian Learning . . .

Maximizing this lower bound, $\mathcal{F}$, leads to **EM-like** updates:

$$\begin{aligned} Q^*_{\mathbf{x}}(\mathbf{x}) &\propto \exp \langle \ln P(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) \rangle_{Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} && E\!-\!like\ step \\ Q^*_{\boldsymbol{\theta}}(\boldsymbol{\theta}) &\propto P(\boldsymbol{\theta}) \exp \langle \ln P(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) \rangle_{Q_{\mathbf{x}}(\mathbf{x})} && M\!-\!like\ step \end{aligned}$$

Maximizing $\mathcal{F}$ is equivalent to minimizing KL-divergence between the *approximate posterior*, $Q(\boldsymbol{\theta})Q(\mathbf{x})$ and the *true posterior*, $P(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$.

Conjugate-Exponential models

Let’s focus on *conjugate-exponential* (**CE**) models, which satisfy **(1)** and **(2)**:
Condition (1). The joint probability over *variables* is in the exponential family:

$$P(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = f(\mathbf{x}, \mathbf{y})\ g(\boldsymbol{\theta}) \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}, \mathbf{y})\right\}$$

where $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the vector of *natural parameters*, $\mathbf{u}$ are *sufficient statistics*
Condition (2). The prior over *parameters* is conjugate to this joint probability:

$$P(\boldsymbol{\theta}|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})\ g(\boldsymbol{\theta})^\eta \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \boldsymbol{\nu}\right\}$$

where η and $\boldsymbol{\nu}$ are hyperparameters of the prior.
Conjugate priors are computationally convenient and have an intuitive interpretation:

- η: number of pseudo-observations
- $\boldsymbol{\nu}$: values of pseudo-observations

Conjugate-Exponential examples

In the **CE** family:

- Gaussian mixtures
- factor analysis, probabilistic PCA
- hidden Markov models and factorial HMMs
- linear dynamical systems and switching models
- discrete-variable belief networks

Other as yet undreamt-of models can combine Gaussian, Gamma, Poisson, Dirichlet, Wishart, Multinomial and others.

Not in the **CE** family:

- Boltzmann machines, MRFs (no conjugacy)
- logistic regression (no conjugacy)
- sigmoid belief networks (not exponential)
- independent components analysis (not exponential)

Note: one can often approximate these models with models in the **CE** family.

The Variational EM algorithm

VE Step: Compute the expected sufficient statistics $\sum_i \overline{\mathbf{u}}(\mathbf{x}_i, \mathbf{y}_i)$ under the hidden variable distributions $Q_{\mathbf{x}_i}(\mathbf{x}_i)$.

VM Step: Compute expected natural parameters $\overline{\phi}(\boldsymbol{\theta})$ under the parameter distribution given by $\tilde{\eta}$ and $\tilde{\boldsymbol{\nu}}$.

Properties:

- Reduces to the EM algorithm if $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$.
- $\mathcal{F}$ increases monotonically, and incorporates the model complexity penalty.
- Analytical parameter distributions (but not constrained to be Gaussian).
- VE step has same complexity as corresponding E step.
- We can use the junction tree, belief propagation, Kalman filter, etc, algorithms in the VE step of VEM, but *using expected natural parameters*.

View 2: Large models

We ought not to limit the complexity of our model a priori (e.g. number of hidden states, number of basis functions, number of mixture components, etc) since we don't believe that the real data was actually generated from a statistical model with a small number of parameters.

Therefore, regardless of how much training data we have, we should consider models with as many parameters as we can handle computationally.

Neal (1994) showed that MLPs with large numbers of hidden units achieved good performance on small data sets. He used MCMC techniques to average over parameters.

Here there is no model order selection task:

- No need to evaluate evidence (which is often difficult).
- We don't need or want to use Occam's razor to limit the number of parameters in our model.

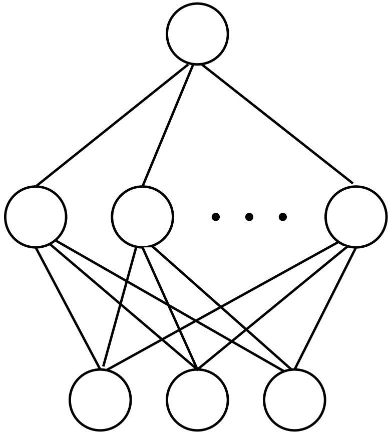
In fact, we may even want to consider doing inference in models with an infinite number of parameters...

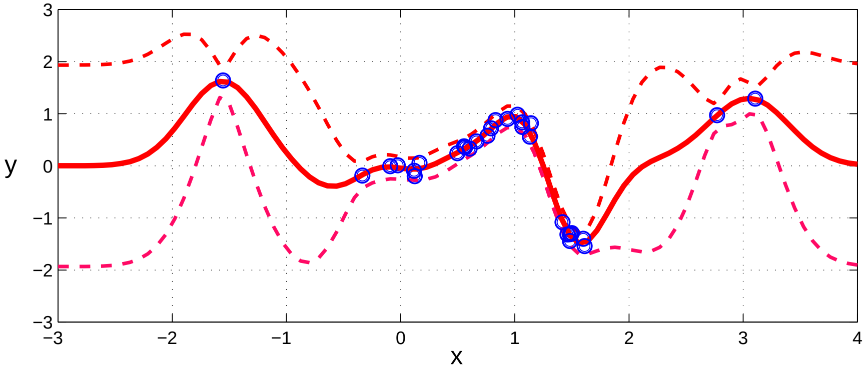
Infinite Models 1: Gaussian Processes

Neal (1994) showed that a one-hidden-layer neural network with bounded activation function and Gaussian prior over the weights and biases converges to a (nonstationary) Gaussian process prior over functions.

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(0, C(\mathbf{x}))$$

where e.g. $C_{ij} \equiv C(x_i, x_j) = g(|x_i - x_j|)$.



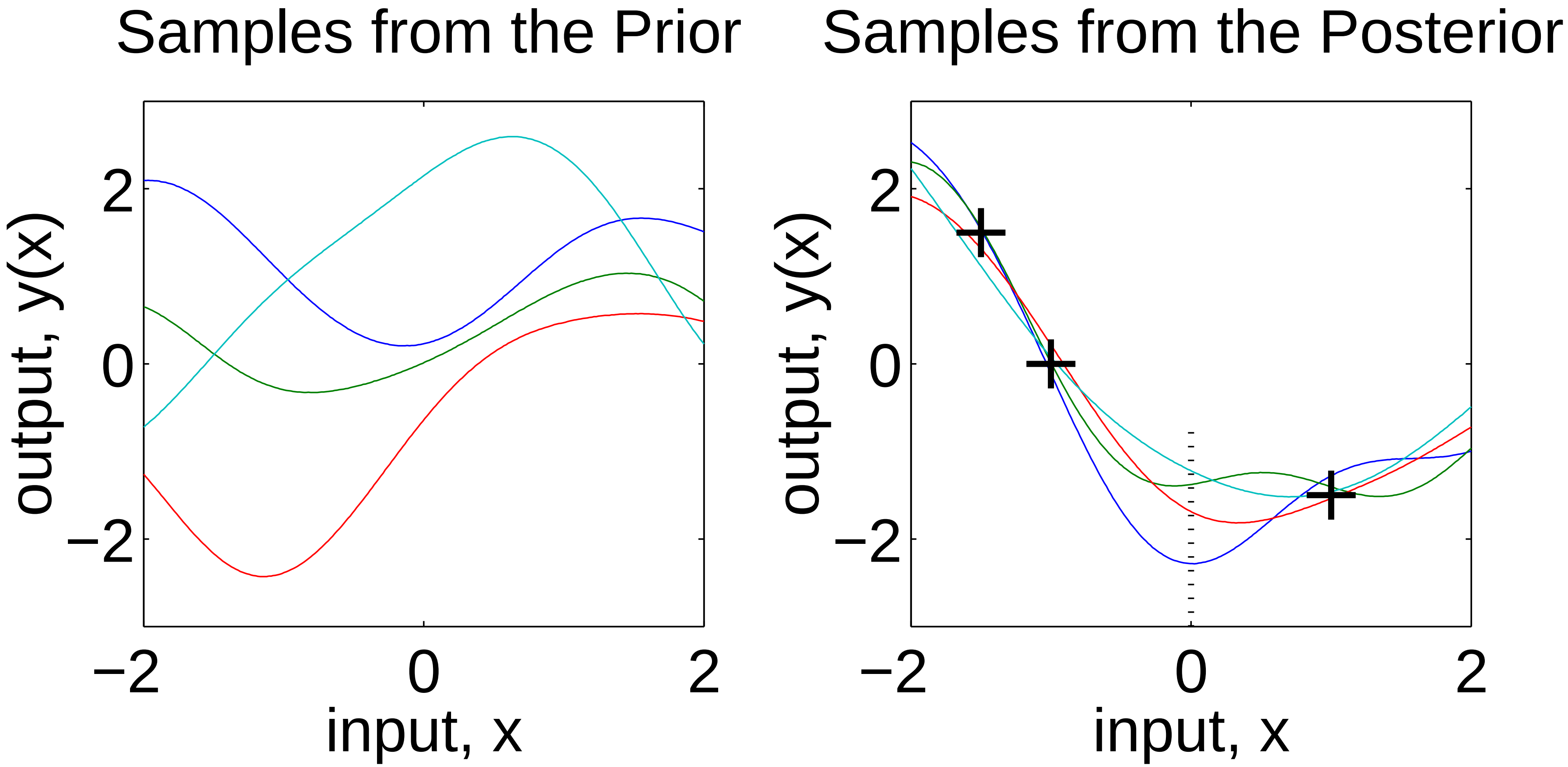


Gaussian Process with Error Bars

Bayesian inference is GPs is conceptually and algorithmically much easier than inference in large neural networks.

Williams (1995; 1996) and Rasmussen (1996) have evaluated GPs as regression models and shown that they are very good.

Gaussian Processes: prior over functions



Linear Regression $\Rightarrow$ Gaussian Processes

in four steps...

1. Linear Regression with inputs $\mathbf{x}_i$ and outputs y_i: $y_i = \sum_k w_k x_{ik} + \epsilon_i$

2. Kernel Linear Regression: $y_i = \sum_k w_k \phi_k(\mathbf{x}_i) + \epsilon_i$

3. Bayesian Kernel Linear Regression:

$$w_k \sim N(0, \beta_k) \quad \text{[indep. of } w_\ell\text{]}, \qquad \epsilon_i \sim N(0, \sigma^2)$$

4. Now, *integrate out* the weights, w_k:

$$\langle y_i \rangle = 0, \qquad \langle y_i y_j \rangle = \sum_k \beta_k \phi_k(\mathbf{x}_i) \phi_k(\mathbf{x}_j) + \delta_{ij}\sigma^2 \equiv C_{ij}$$

This is a Gaussian process with covariance function:

$$C(\mathbf{x}, \mathbf{x}') = \sum_k \beta_k \phi_k(\mathbf{x}) \phi_k(\mathbf{x}') + \delta_{ij}\sigma^2 \equiv C_{ij}$$

This is a Gaussian process with finite number of basis functions. Many useful GP covariance functions correspond to infinitely many kernels.

Infinite Models 2: Infinite Gaussian Mixtures

Following Neal (1991), Rasmussen (2000) showed that it is possible to do inference in countably infinite mixtures of Gaussians.

$$\begin{aligned} P(x_1, \ldots, x_N | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{N} \sum_{j=1}^{K} \pi_j \, \mathcal{N}(x_i | \mu_j, \Sigma_j) \\ = \sum_{\mathbf{s}} P(\mathbf{s}, \mathbf{x} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{\mathbf{s}} \prod_{i=1}^{N} \prod_{j=1}^{K} [\pi_j \, \mathcal{N}(x_i | \mu_j, \Sigma_j)]^{\delta(s_i, j)} \end{aligned}$$

Joint distribution of indicators is **multinomial**

$$P(s_1, \ldots, s_N | \boldsymbol{\pi}) = \prod_{j=1}^{K} \pi_j^{n_j}, \qquad n_j = \sum_{i=1}^{N} \delta(s_i, j) \ .$$

Mixing proportions are given symmetric Dirichlet **prior**

$$P(\boldsymbol{\pi} | \beta) = \frac{\Gamma(\beta)}{\Gamma(\beta/K)^K} \prod_{j=1}^{K} \pi_j^{\beta/K - 1}$$

Infinite Gaussian Mixtures (continued)

Joint distribution of indicators is **multinomial**

$$P(s_1, \ldots, s_N | \boldsymbol{\pi}) = \prod_{j=1}^{K} \pi_j^{n_j}, \qquad n_j = \sum_{i=1}^{N} \delta(s_i, j) \ .$$

Mixing proportions are given symmetric Dirichlet **conjugate prior**

$$P(\boldsymbol{\pi} | \beta) = \frac{\Gamma(\beta)}{\Gamma(\beta/K)^K} \prod_{j=1}^{K} \pi_j^{\beta/K-1}$$

Integrating out the mixing proportions we obtain

$$P(s_1, \ldots, s_N | \beta) = \int d\boldsymbol{\pi} \ P(s_1, \ldots, s_N | \boldsymbol{\pi}) P(\boldsymbol{\pi} | \beta) = \frac{\Gamma(\beta)}{\Gamma(n + \beta)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \beta/K)}{\Gamma(\beta/K)}$$

This yields a Dirichlet Process over indicator variables.

Dirichlet Process Conditional Probabilities

Conditional Probabilities: Finite K

$$P(s_i = j | \mathbf{s}_{-i}, \beta) = \frac{n_{-i,j} + \beta/K}{N - 1 + \beta}$$

where $\mathbf{s}_{-i}$ denotes all indices except i, and $n_{-i,j}$ is total number of observations of indicator j excluding i^{th}.
DP: more populous classes are more more likely to be joined

Conditional Probabilities: Infinite K

Taking the limit as $K \to \infty$ yields the conditionals

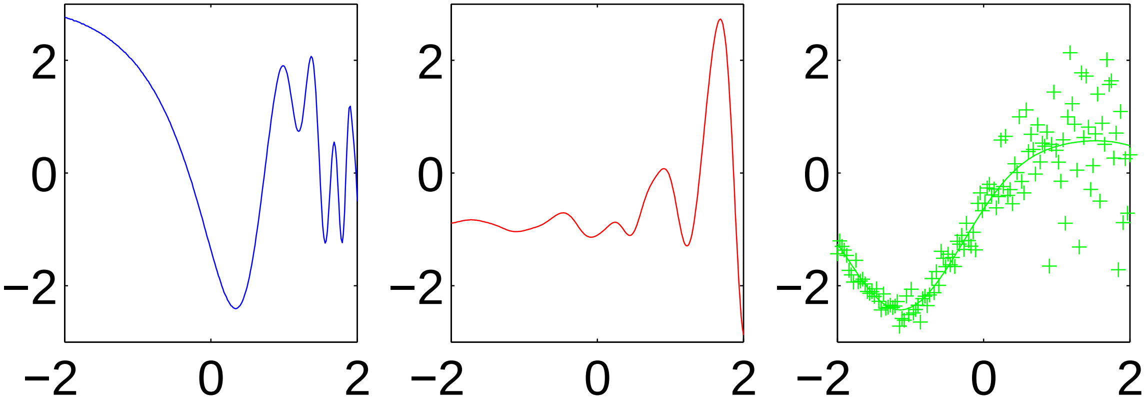
$$P(s_i = j | \mathbf{s}_{-i}, \beta) = \begin{cases} \frac{n_{-i,j}}{N-1+\beta} & j \text{ represented} \\ \frac{\beta}{N-1+\beta} & \text{all } j \text{ not represented} \end{cases}$$

Left over mass, β, $\Rightarrow$ **countably infinite** number of indicator settings.
Gibbs sampling from posterior of indicators is easy!

Infinite Models 3: Infinite Mixtures of Experts

Motivation:

1. Difficult to specify flexible GP covariance structures:



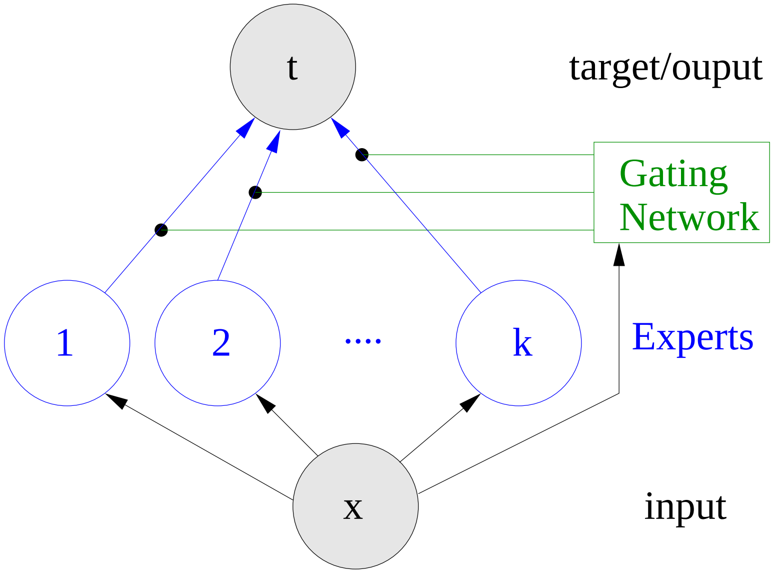
eg, varying spatial frequency, varying signal amplitude, varying noise etc.

2. Predictions and training requires C^{-1} which has $\mathcal{O}(n^3)$ complexity.

Solution: the divide and conquer strategy of Mixture of Experts.
A (countably infinite) mixture of Gaussian Processes, allows:

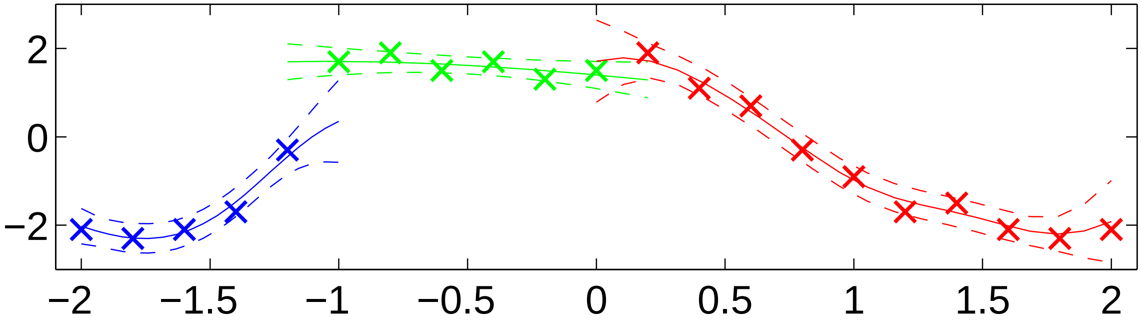
- different covariance functions in different parts of space
- divide-and-conquer efficiency (by splitting $\mathcal{O}(n^3)$ between experts).

Mixture of Experts Review



Simultaneously train the gating network and the experts using the likelihood:

$$p(\mathbf{t}|\mathbf{x}, \Psi, w) = \prod_{i=1}^{n} \sum_{j=1}^{k} p(c_i = j|x^{(i)}, w) p(t^{(i)}|c_i = j, x^{(i)}, \Psi_j).$$

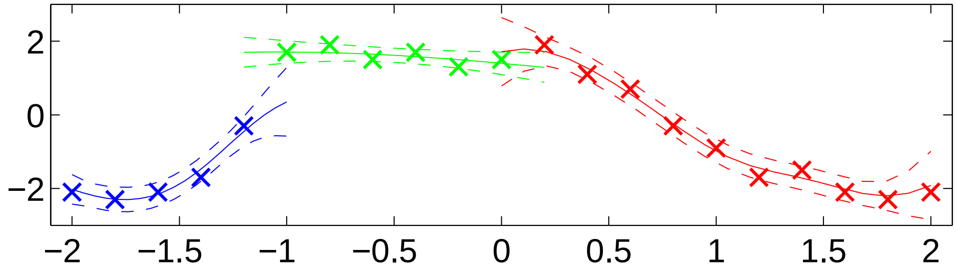


Mixture of GP Experts

The likelihood traditionally used for Mixture of Experts:

$$p(\mathbf{t}|\mathbf{x}, \Psi, w) = \prod_{i=1}^{n} \sum_{j=1}^{k} p(c_i = j|x^{(i)}, w) p(t^{(i)}|c_i = j, x^{(i)}, \Psi_j),$$

assumes the data is iid given the experts.
This does not hold for GPs: The experts change depending on what other examples are assigned to them:



The likelihood becomes a sum over (exponentially many) possible assignments:

$$p(\mathbf{t}|\mathbf{x}, \Psi, w) = \sum_{\mathbf{c}} p(\mathbf{c}|\mathbf{x}, w) \prod_{j=1}^{k} p(\{t^{(i)} : c_i = j\}|\mathbf{x}, \Psi_j).$$

Gating Network: Input-dependent Dirichlet Process

Usual Dirichlet Process:

$$P(c_i = j|\mathbf{c}_{-i}, \beta) = \begin{cases} \frac{n_{-i,j}}{N-1+\beta} & j \text{ represented} \\ \frac{\beta}{N-1+\beta} & \text{all } j \text{ not represented} \end{cases}$$

Input-Dependent Dirichlet Process:

$$P(c_i = j|\mathbf{c}_{-i}, \mathbf{x}, \beta, w) = \begin{cases} \frac{\tilde{n}_{-i,j}(\mathbf{x})}{N-1+\beta} & j \text{ represented} \\ \frac{\beta}{N-1+\beta} & \text{all } j \text{ not represented} \end{cases}$$

where the gating function gives a “local estimate” of the occupation number:

$$\tilde{n}_{-i,j}(\mathbf{x}) = (N-1)P(c_i = j|c_{-i}, \mathbf{x}, w),$$

Bayesian inference in the model

Using ideas of Gibbs sampling, we can alternately:

1) Update the parameters given the indicators:

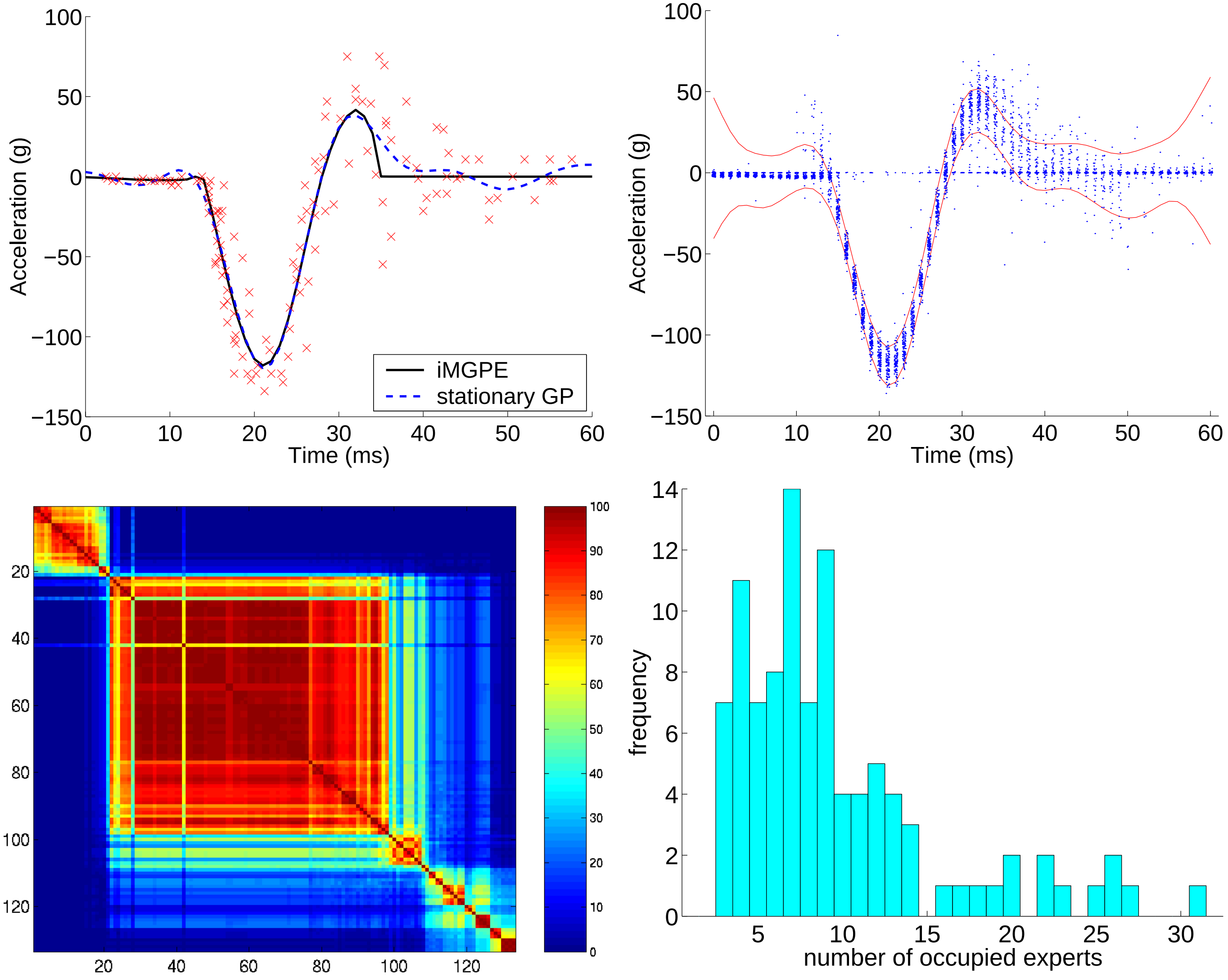
   - GP hyperparameters are sampled by Hybrid Monte Carlo
   - gating function kernel widths are sampled with Metropolis

2) Update the indicators given the parameters:

   - Sequentially Gibbs sample the indicators combining the gating $p(c_i|c_{-i}, \mathbf{x}, w)$ and expert $p(t_i|c_i, \mathbf{x}, \Psi)$ information

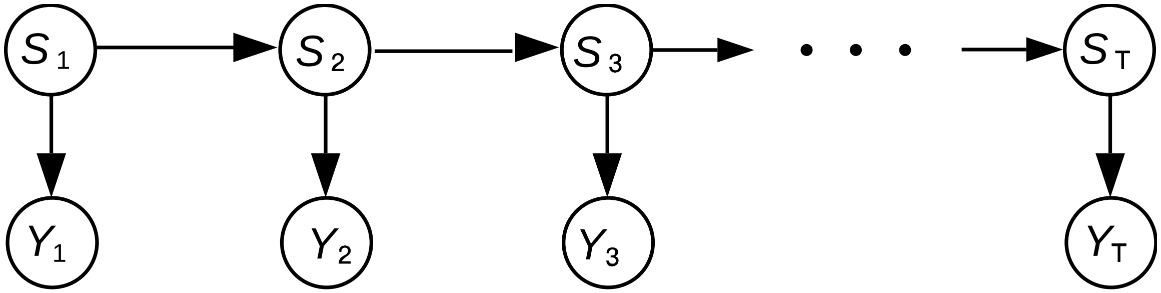
Complexity can be further reduced by constraining $n_j < n_{\max}$.

Infinite Mixtures of Experts Results



(Rasmussen and Ghahramani, 2001)

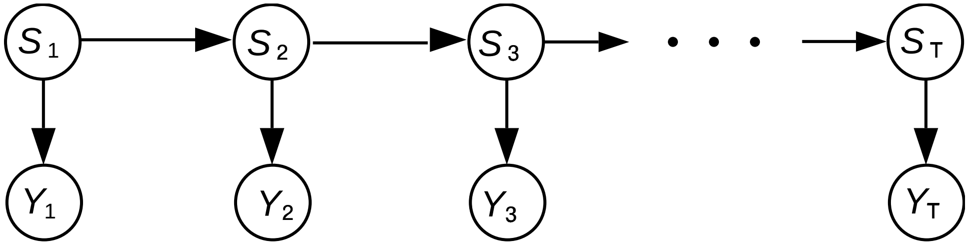
Infinite Models 4: Infinite hidden Markov Models



Motivation: We want to model data with HMMs without worrying about overfitting, picking number of states, picking architectures...

Review of Hidden Markov Models (HMMs)

Generative graphical model: **hidden states** s_t, **emitted symbols** y_t

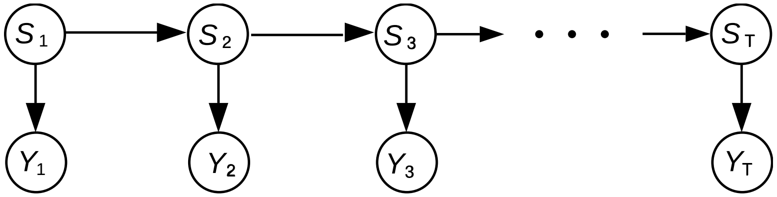


Hidden state evolves as a Markov process

$$P(s_{1:T}|A) = P(s_1|\boldsymbol{\pi}_0) \prod_{t=1}^{T-1} P(s_{t+1}|s_t) \ , \qquad \begin{array}{c} P(s_{t+1} = j|s_t = i) = A_{ij} \\ i, j \in \{1, \ldots, K\} \ . \end{array}$$

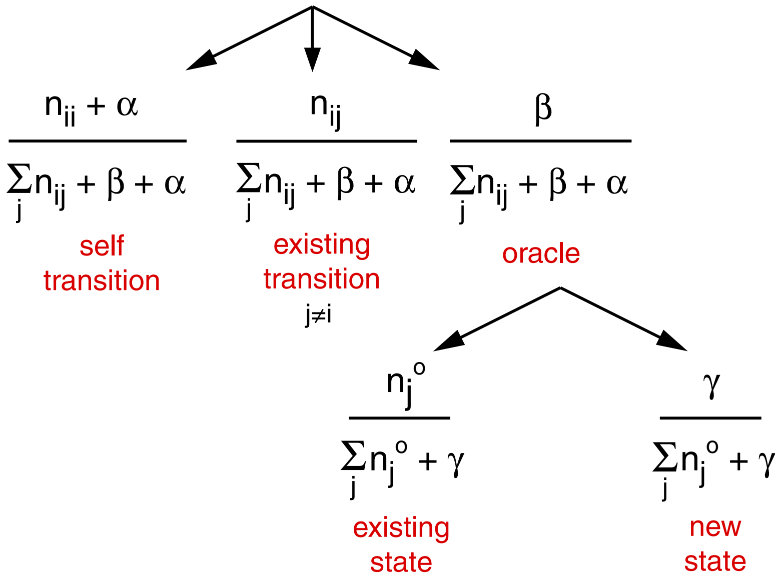
Observation model e.g. **discrete** y_t symbols from an alphabet produced according to an emission matrix, $P(y_t = \ell|s_t = i) = E_{i\ell}$.

Infinite HMMs



Approach: Countably-infinite hidden states. Deal with both transition and emission processes using a two-level hierarchical Dirichlet process.

Transition process



$$\frac{n_{ii}+\alpha}{\sum_j n_{ij}+\beta+\alpha}$$

self transition

$$\frac{n_{ij}}{\sum_j n_{ij}+\beta+\alpha}$$

existing transition

j≠i

$$\frac{\beta}{\sum_j n_{ij}+\beta+\alpha}$$

oracle

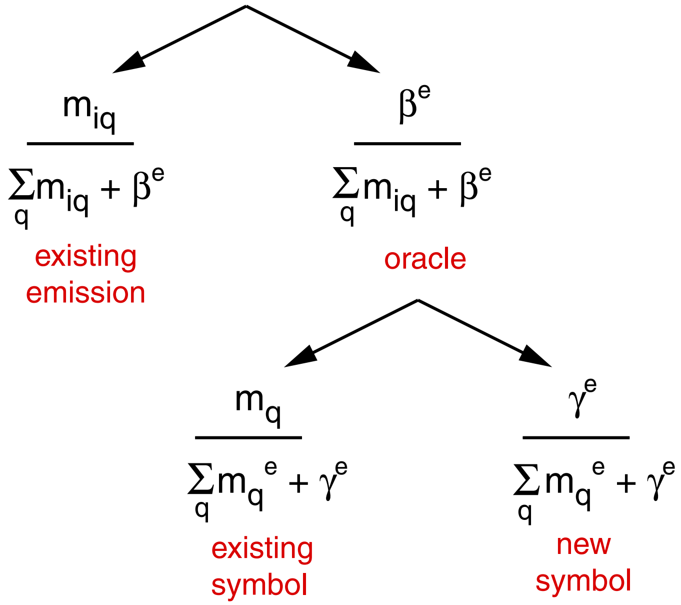
$$\frac{n_j^o}{\sum_j n_j^o+\gamma}$$

existing state

$$\frac{\gamma}{\sum_j n_j^o+\gamma}$$

new state

Emission process



$$\frac{m_{iq}}{\sum_q m_{iq}+\beta^e}$$

existing emission

$$\frac{\beta^e}{\sum_q m_{iq}+\beta^e}$$

oracle

$$\frac{m_q}{\sum_q m_q^e+\gamma^e}$$

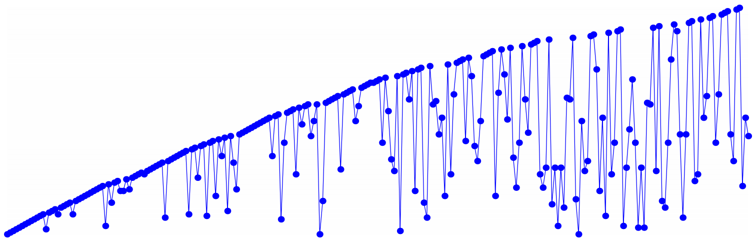
existing symbol

$$\frac{\gamma^e}{\sum_q m_q^e+\gamma^e}$$

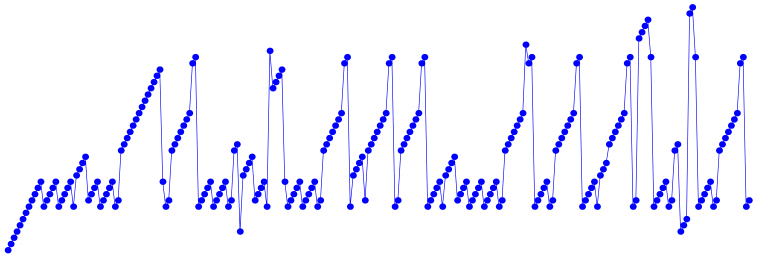
new symbol

Gibbs sampling over the states is possible, while all parameters are implicitly integrated out; only five hyperparameters need to be inferred (Beal, Ghahramani, and Rasmussen, 2001).

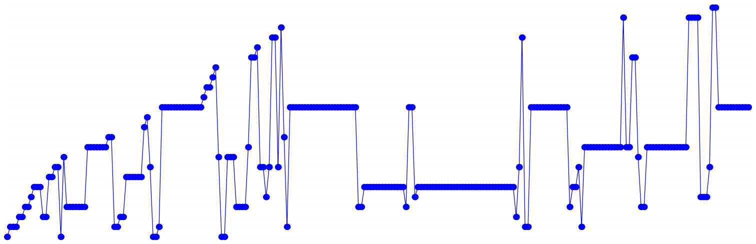
Trajectories under the Prior



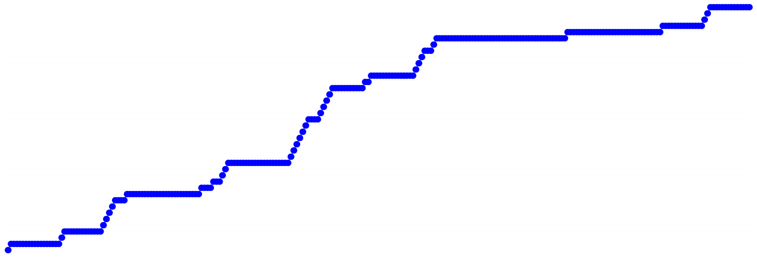
explorative: $\alpha = 0.1$, $\beta = 1000$, $\gamma = 100$



repetitive: $\alpha = 0$, $\beta = 0.1$, $\gamma = 100$



self-transitioning: $\alpha = 2$, $\beta = 2$, $\gamma = 20$



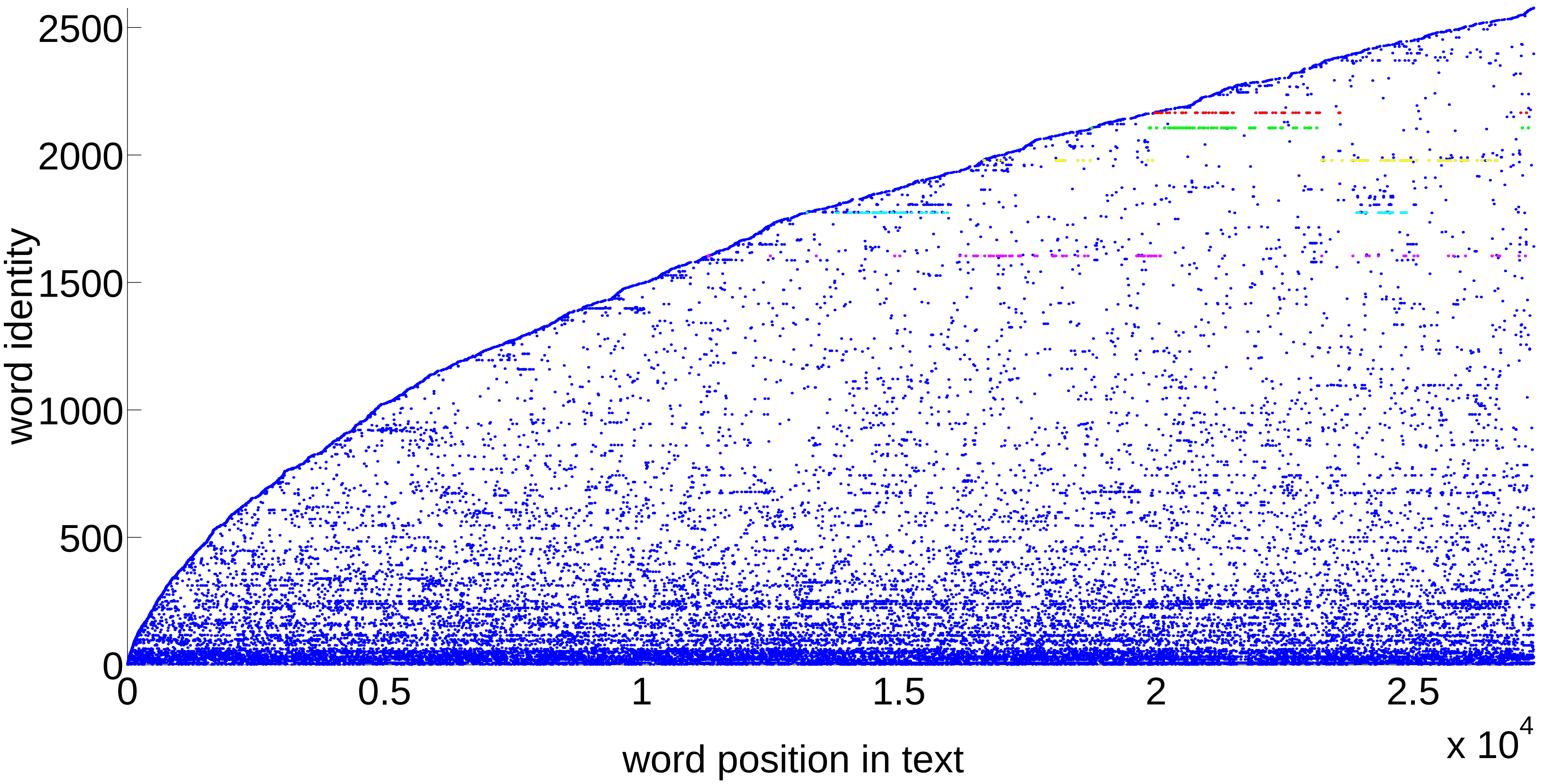
ramping: $\alpha = 1$, $\beta = 1$, $\gamma = 10000$

Just 3 hyperparameters provide:

- slow/fast dynamics (α)
- sparse/dense transition matrices (β)
- many/few states (γ)
- left→right structure, with multiple interacting cycles

Real data

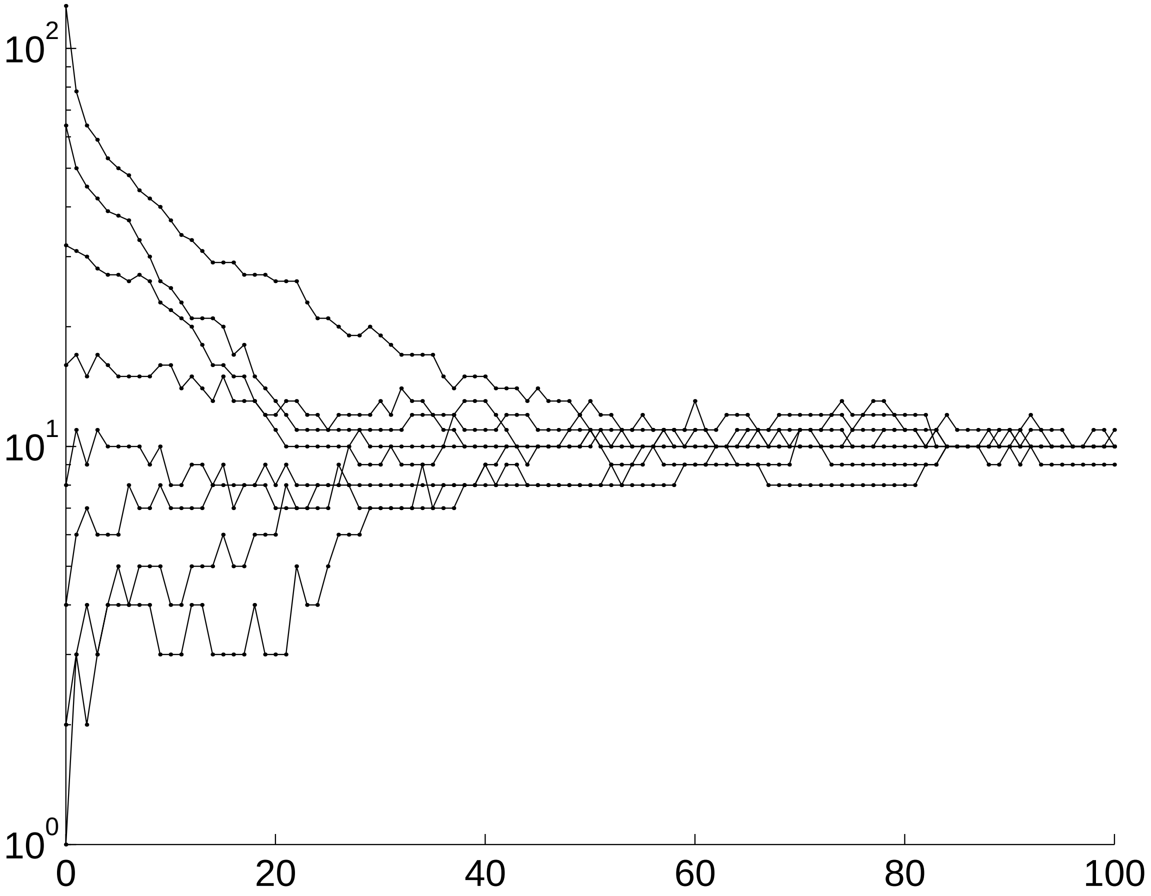
Lewis Carroll's *Alice's Adventures in Wonderland*



With a *finite* alphabet a model would assign zero likelihood to a test sequence containing any symbols not present in the training set(s).
In iHMMs, at each time step the hidden state s_t emits a symbol y_t, which can possibly come from an *infinite* alphabet.

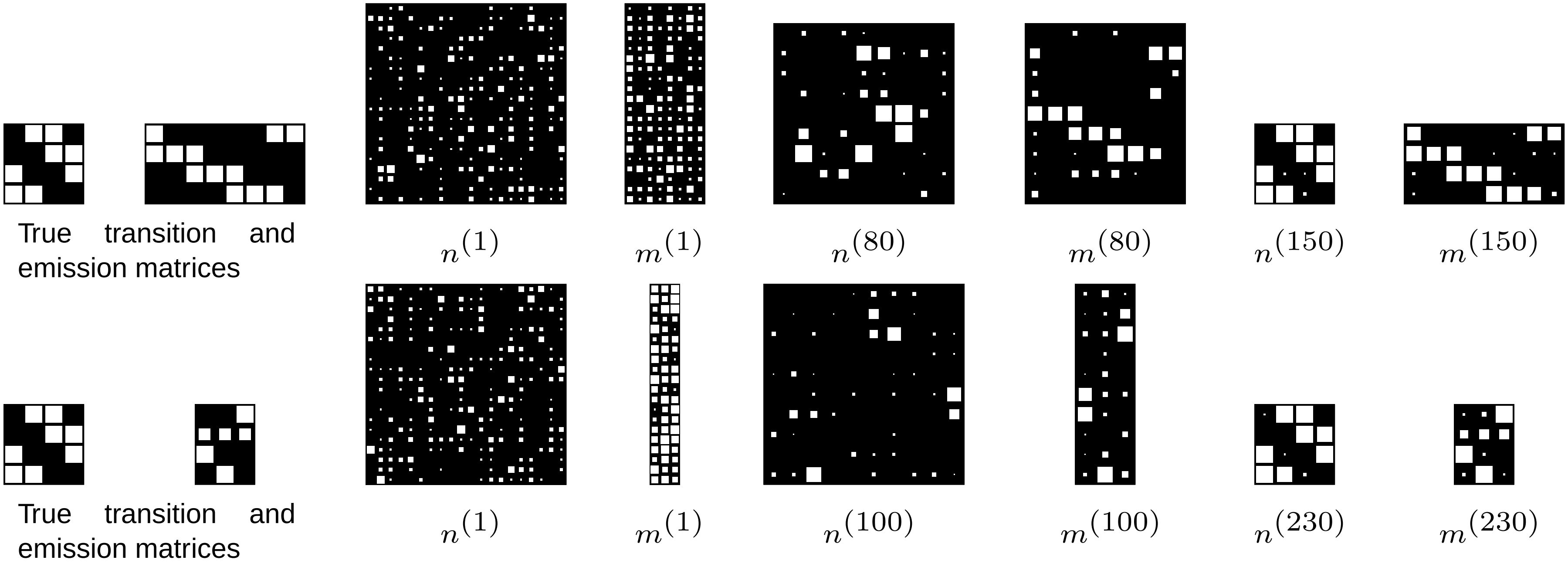
A toy example

ABCDEFEDCBABCDEFEDCBABCDEFEDCBABCDEFEDCB...
This requires minimally 10 states to capture.



Number of represented states vs Gibbs sweeps

iHMM Results



True transition and emission matrices

$n^{(1)}$ $m^{(1)}$ $n^{(80)}$ $m^{(80)}$ $n^{(150)}$ $m^{(150)}$

True transition and emission matrices

$n^{(1)}$ $m^{(1)}$ $n^{(100)}$ $m^{(100)}$ $n^{(230)}$ $m^{(230)}$

True and learned transition and emission probabilities/count matrices up to permutation of the hidden states; lighter boxes correspond to higher values.

(top row) Expansive HMM. Count matrix pairs $\{n, m\}$ are displayed after $\{1, 80, 150\}$ sweeps of Gibbs sampling.

(bottom row) Compressive HMM. Similar to top row displaying count matrices after $\{1, 100, 230\}$ sweeps of Gibbs sampling.

See hmm2.avi and hmm3.avi

Alice Results

- Trained on 1st chapter (10787 characters: A . . . Z, $\langle$space$\rangle$, $\langle$period$\rangle$) =2046 words.
- iHMM initialized with random sequence of 30 states. $\alpha = 0; \beta = \beta^e = \gamma = \gamma^e = 1$.
- 1000 Gibbs sweeps (=several hours in Matlab).
- n matrix starts out full, ends up sparse (14% full).

200 character fantasies...

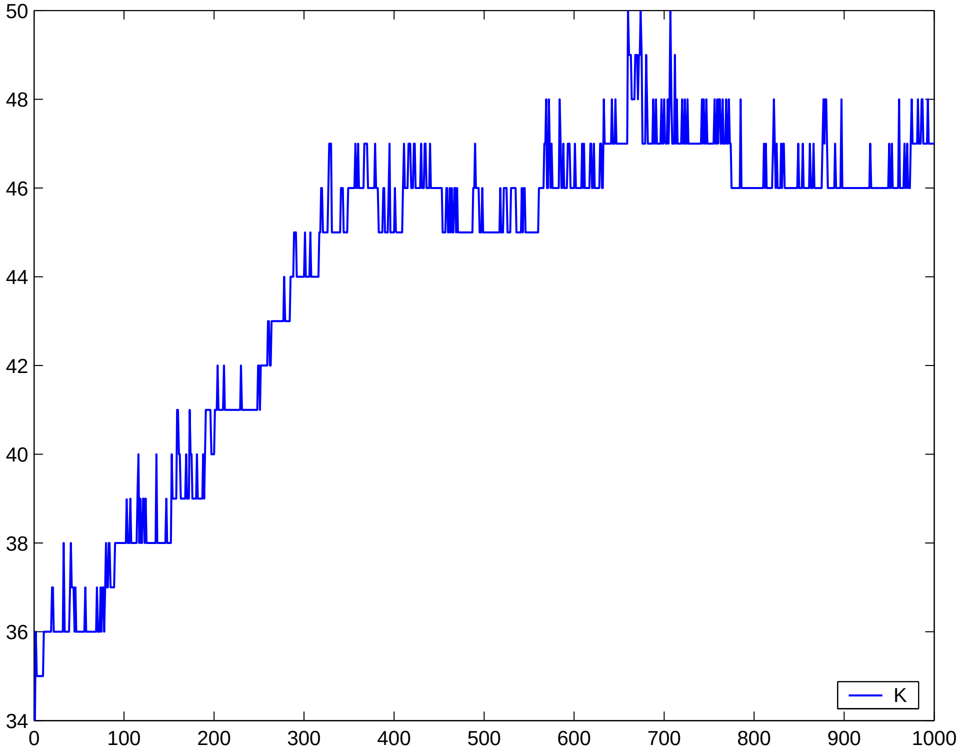
1: LTEMFAODEOHADONARNL SAE UDSEN DTTET ETIR NM H VEDEIPH L.SHYIPFADB OMEBEGLSENTEI GN HEOWDA EELE HEFOMADEE IS AL THWRR KH TDAAAC CHDEE OIGW OHRBOOLEODT DSECT M OEDPGTYHIHNOL CAEGTR.ROHA NOHTR.L

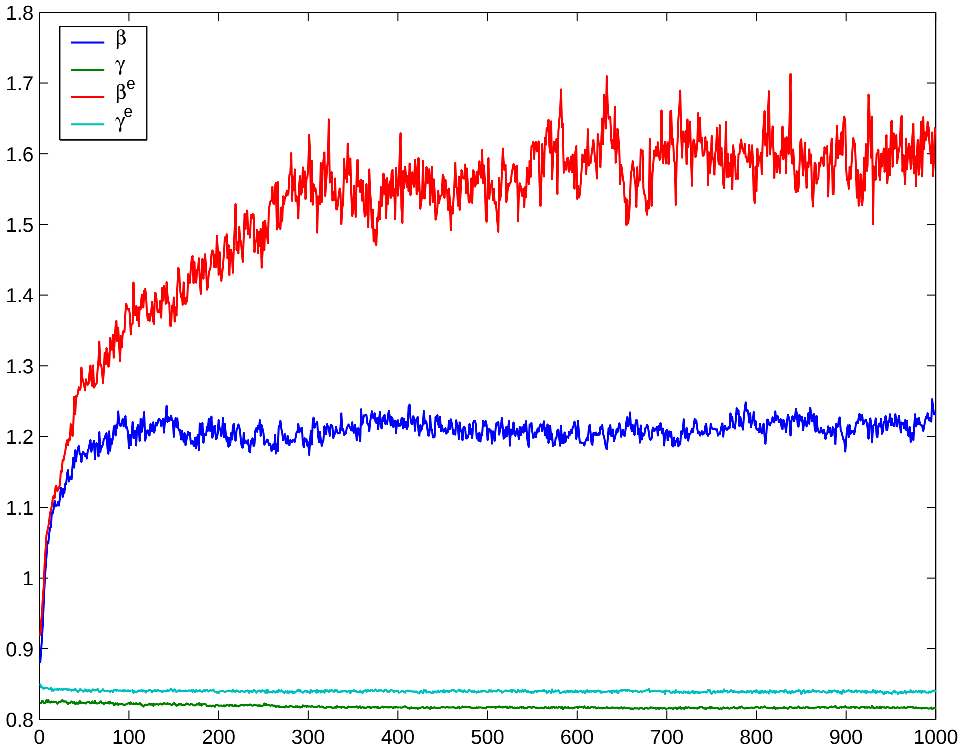
250: AREDIND DUW THE JEDING THE BUBLE MER.FION SO COR.THAN THALD THE BATHERSTHWE ICE WARLVE I TOMEHEDS I LISHT LAKT ORTH.A CEUT.INY OBER.GERD POR GRIEN THE THIS FICE HIGE TO SO.A REMELDLE THEN.SHILD TACE G

500: H ON ULY KER IN WHINGLE THICHEY TEIND EARFINK THATH IN ATS GOAP AT.FO ANICES IN RELL A GOR ARGOR PEN EUGUGTTHT ON THIND NOW BE WIT OR ANND YOADE WAS FOUE CAIT DOND SEAS HAMBER ANK THINK ME.HES URNDEY

1000: BUT.THOUGHT ANGERE SHERE ACRAT OR WASS WILE DOOF SHE.WAS ABBORE GLEAT DOING ALIRE AT TOO AMIMESSOF ON SHAM LUZDERY AMALT ANDING A BUPLA BUT THE LIDTIND BEKER HAGE FEMESETIMEY BUT NOTE GD I SO CALL OVE

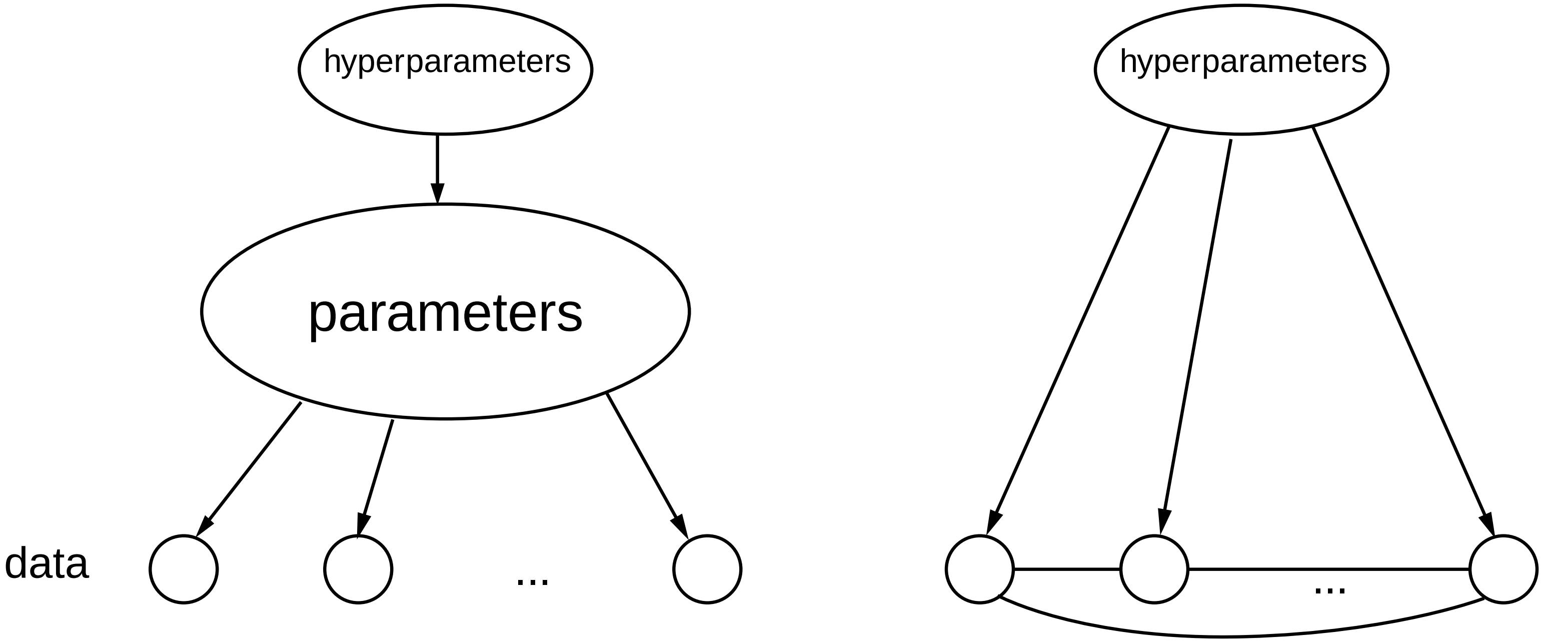
Alice Results: Number of States and Hyperparameters





Which view, 1 or 2?

In theory, view 2 (large/infinite models) is more natural and preferable. But models become nonparametric and often require sampling or $\mathcal{O}(n^3)$ computations (e.g. GPs).



In practice, view 1 (occam's razor) is sometimes attractive, yielding smaller models and allowing deterministic (e.g. variational) approximation methods.

Summary & Conclusions

- Bayesian learning avoids overfitting and can be used to do model selection.

- Two views: model selection via Occam's Razor, versus large/infinite models.

- View 1 - a practical approach: variational approximations
  - Variational EM for CE models and propagation algorithms

- View 2 - Gaussian processes, infinite mixtures, mixture of experts & HMMs.
  - Results in non-parametric models, often requires sampling.

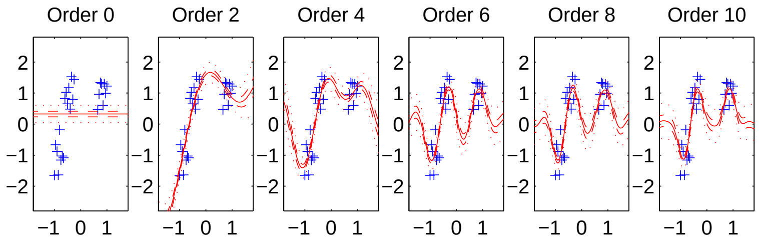
- In the limit of small amounts of data, we don't necessarily favour small models — rather the posterior over model orders becomes flat.

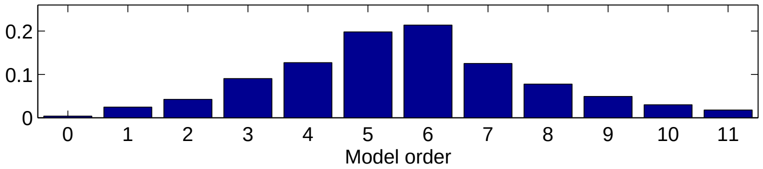
- The two views can be reconciled in the following way: Model complexity $\neq$ number of parameters, Occam's razor can still work selecting between different infinite models (e.g. rough vs smooth GPs).

Scaling the parameter priors

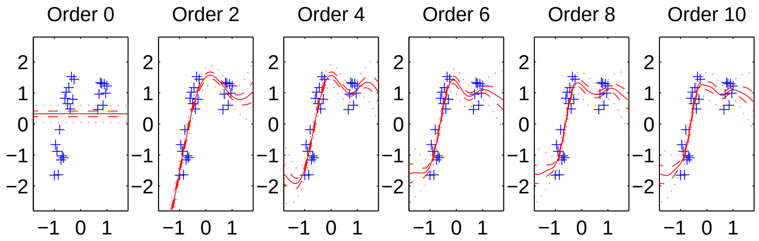
To implement each view it is essential to scale parameter priors appropriately — this determines whether an Occam’s hill is present or not.

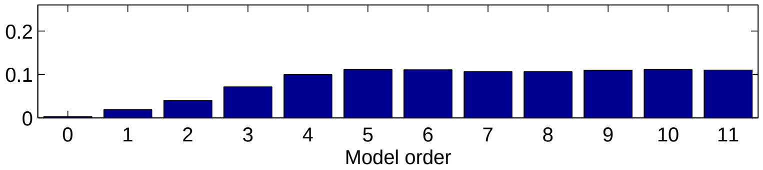
Unscaled models:





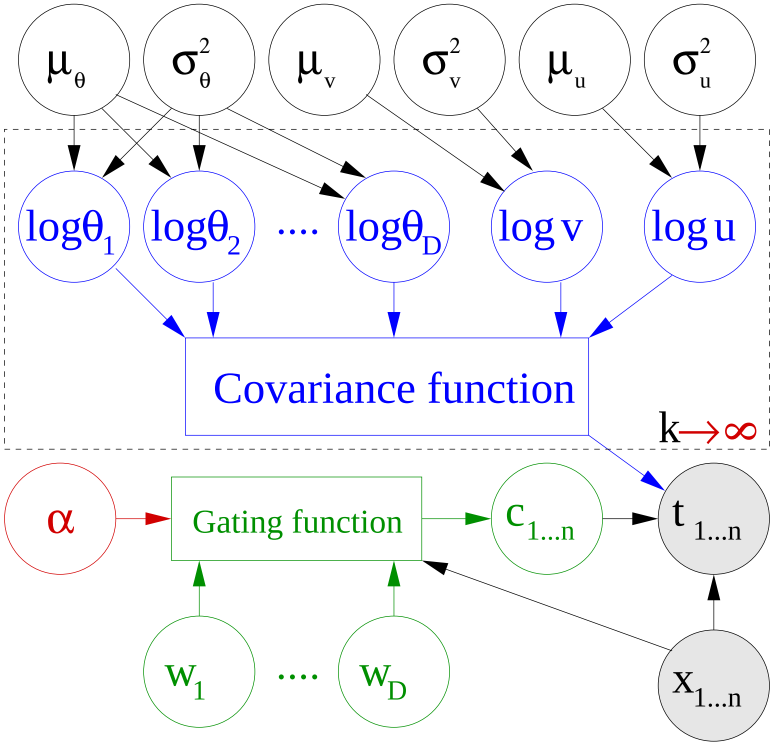
Scaled models:





Appendix: Infinite Mixture of Experts

Graphical Model for iMGPE



$x_{1\dots n}$, $t_{1\dots n}$	inputs and targets (observed)
$c_{1\dots n}$	indicators $c_i \in \{1 \dots k\}$
w	gating function kernel widths
$\Psi = \{\theta, v, u\}$	GP hyperparameters: θ input length scales v signal variance u noise variance
α	the Dirichlet process concentration parameter
μ's, σ^2's	GP hyper-hypers

How Many Experts?

simple, assume an infinite number of experts!

Dirichlet Process with concentration parameter α defines the conditional prior for an indicator to be:

$$p(c_i = j | c_{-i}, \alpha) = \frac{n_{-i,j}}{n - 1 + \alpha}$$

where $n_{-i,j}$ is the *occupation number* for expert j (excluding example i) for currently occupied experts.

The total probability of all (infinitely many) unoccupied experts combined:

$$p(c_i = j_{\text{new}} | c_{-i}, \alpha) = \frac{\alpha}{n - 1 + \alpha}$$

Input-Dependent Dirichlet Process combines the DP with a gating function:

$$\tilde{n}_{-i,j} = (n-1) p(c_i = j | c_{-i}, \mathbf{x}, w),$$

which gives a “local estimate” of the occupation number.

The algorithm

Sample:

1. do a Gibbs sampling sweep over all indicators

2. sample gating function kernel widths w using Metropolis

3. for each of the occupied experts:
   do Hybrid Monte Carlo for the GP hyperparameters θ, v, u.

4. Sample the Dirichlet process concentration parameter, α using Adaptive Rejection Sampling.

5. Optimize the GP hyper-hypers, μ, σ^2.

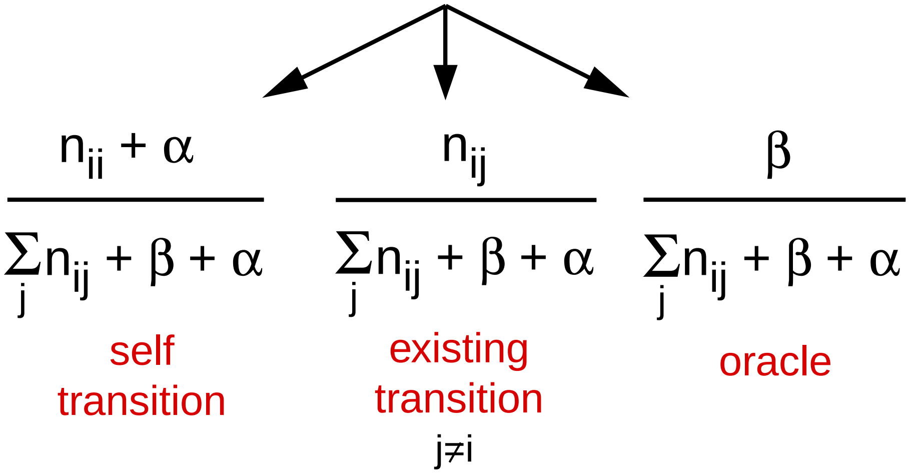
Repeat until the Markov chain has adequately sampled the posterior.

Appendix: Infinite HMMs

Generative model for hidden state

Propose transition to s_{t+1} conditional on current state, s_t.
Existing transitions are more probable,
thus giving rise to *typical trajectories*.

$$n_{ij} = \begin{array}{cc} & s_{t+1} \longrightarrow \\ \begin{array}{c} s_t \\ \downarrow \end{array} & \left[\begin{array}{cccc} 3 & 17 & 0 & 14 \\ 19 & 2 & 7 & 0 \\ 0 & 3 & 1 & 8 \\ 11 & 7 & 4 & 3 \end{array}\right] \begin{array}{c} \beta \\ \beta \\ \beta \\ \beta \end{array} \end{array}$$



$$\frac{n_{ii} + \alpha}{\sum_j n_{ij} + \beta + \alpha}$$

self transition

$$\frac{n_{ij}}{\sum_j n_{ij} + \beta + \alpha}$$

existing transition

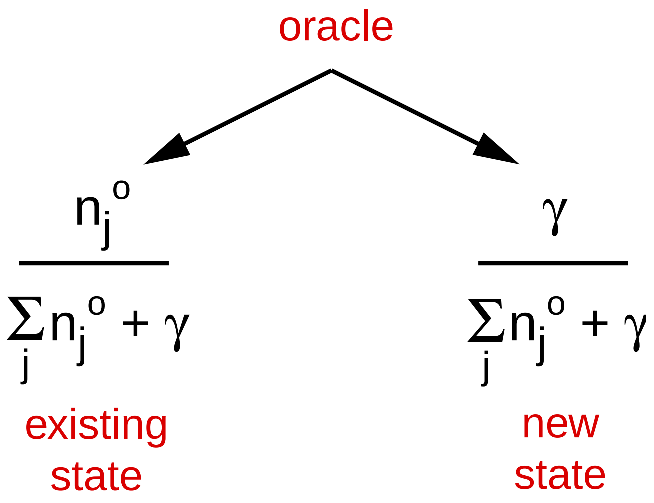
$j \neq i$

$$\frac{\beta}{\sum_j n_{ij} + \beta + \alpha}$$

oracle

If oracle propose according to occupancies.
Previously chosen oracle states are more probable.

$$n_j^o = \begin{array}{c} s_{t+1} \longrightarrow \\ \left[\begin{array}{cccc} 4 & 0 & 9 & 11 \end{array}\right] \ \gamma \end{array}$$



oracle

$$\frac{n_j^o}{\sum_j n_j^o + \gamma}$$

existing state

$$\frac{\gamma}{\sum_j n_j^o + \gamma}$$

new state

Some References

1. Attias H. (1999) Inferring parameters and structure of latent variable models by variational Bayes. Proc. 15th Conference on Uncertainty in Artificial Intelligence.
2. Barber D., Bishop C. M., (1998) Ensemble Learning for MultiLayer Networks. Advances in Neural Information Processing Systems 10..
3. Bishop, C. M. (1999). Variational principal components. Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99 (pp. 509–514).
4. Beal, M. J., Ghahramani, Z. and Rasmussen, C. E. (2001) The Infinite Hidden Markov Model. To appear in NIPS2001.
5. Ghahramani, Z. and Beal, M.J. (1999) Variational inference for Bayesian mixtures of factor analysers. In Neural Information Processing Systems 12.
6. Ghahramani, Z. and Beal, M.J. (2000) Propagation algorithms for variational Bayesian learning. In Neural Information Processing Systems 13
7. Hinton, G. E., and van Camp, D. (1993) Keeping neural networks simple by minimizing the description length of the weights. In Proc. 6th Annu. Workshop on Comput. Learning Theory , pp. 5–13. ACM Press, New York, NY.
8. MacKay, D. J. C. (1995) Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6: 469–505.
9. Miskin J. and D. J. C. MacKay, Ensemble learning independent component analysis for blind separation and deconvolution of images, in Advances in Independent Component Analysis, M. Girolami, ed., pp. 123–141, Springer, Berlin, 2000.
10. Neal, R. M. (1991) Bayesian mixture modeling by Monte Carlo simulation, Technical Report CRG-TR-91-2, Dept. of Computer Science, University of Toronto, 23 pages.
11. Neal, R. M. (1994) Priors for infinite networks, Technical Report CRG-TR-94-1, Dept. of Computer Science, University of Toronto, 22 pages.

12. Rasmussen, C. E. (1996) Evaluation of Gaussian Processes and other Methods for Non-Linear Regression. Ph.D. thesis, Graduate Department of Computer Science, University of Toronto.

13. Rasmussen, C. E. (1999) The Infinite Gaussian Mixture Model. Advances in Neural Information Processing Systems 12, S.A. Solla, T.K. Leen and K.-R. Mller (eds.), pp. 554-560, MIT Press (2000).

14. Rasmussen, C. E and Ghahramani, Z. (2000) Occam's Razor. Advances in Neural Information Systems 13, MIT Press (2001).

15. Rasmussen, C. E and Ghahramani, Z. (2001) Infinite Mixtures of Gaussian Process Experts. In NIPS2001.

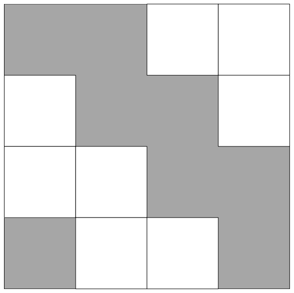
16. Ueda, N. and Ghahramani, Z. (2000) Optimal model inference for Bayesian mixtures of experts. IEEE Neural Networks for Signal Processing. Sydney, Australia.

17. Waterhouse, S., MacKay, D.J.C. & Robinson, T. (1996). Bayesian methods for mixtures of experts. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8. Cambridge, MA: MIT Press.

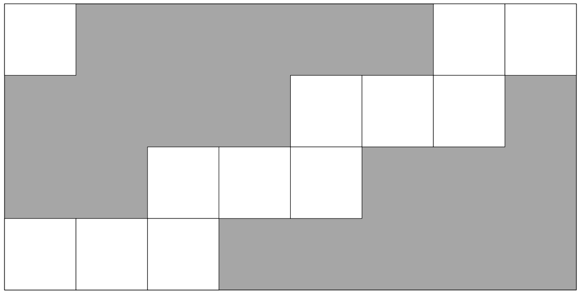
18. Williams, C. K. I., and Rasmussen, C. E. (1996) Gaussian processes for regression. In Advances in Neural Information Processing Systems 8 , ed. by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo.

Another toy example:

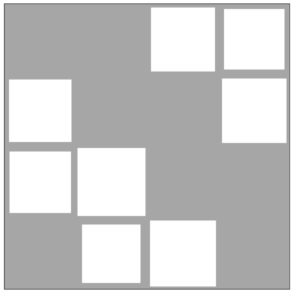
Small HMMs with left-right dynamics:



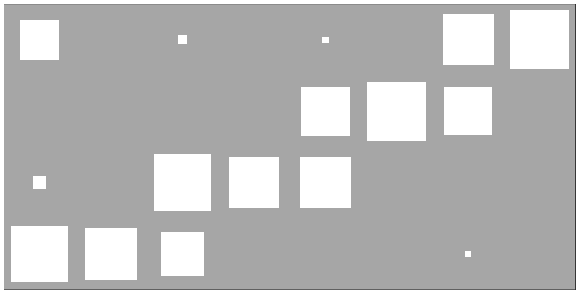
True Transition Matrix



True Emission Matrix



Inferred Transition Counts



Inferred Emission Counts

Sequence of length 800, starting with 20 states, 150 Gibbs sweeps.